REMARKS

Claims 1, 7, 8, 14 and 15 have been amended by this response after final rejection of the claims. Applicants respectfully request that the Examiner enter claims 16 – 17 into the application and consider them as well.

The Examiner again rejected claims 1, 2, 5, 6, 7, 8, 9, 12, 13, 14 and 15 under 35 U.S.C. § 103(a) as being unpatentable over Barnstijn, et al., U.S. Patent No. 5,600,790, further in view of Ogata, et al., U.S. Patent No. 5,758,124. This rejection is again respectfully traversed.

In response to the Applicants' arguments filed June 16, 2000, the Examiner stated that Applicants' arguments have been fully considered but are not persuasive. With respect to claim 1, Applicants contend that Barnstijn, et al. fail to disclose a system where all testing and development occur on the same system. The Examiner relies on Barnstijn, et al. for a teaching that both development and testing is done on a host system. The material referenced by the Examiner does in fact state that the Barnstijn, et al. invention provides a method and system whereby a program intended for a target system can be developed and tested directly on the host system. It is Applicants' position that that statement was taken out of context because it is immediately followed with the following statement: "This approach provides a developer using the invention to observe the operation of the program in the target system as the program is being developed." (Col. 2, ll. 51 – 56). Barnstijn, et al. further teaches at column 3, lines 22 – 47, that "both input and output events are physically executed or initiated using the target system's hardware while one or more applications under development reside and are executed in the host computer development environment. This indicates a teaching that, although the application development is being done a host system, it is using actual devices on a target system through a communications link to test that application. This is further evidenced in claim 1, step (g) where

the claim recites the step of: "executing, at the target system, the target operating system calls, thereby enabling an interaction with the target input/output devices while the trial application program is tested on said host system". To further clarify this distinction between the present invention and Barnstijn, et al., claim 1 of the present invention has been amended in the preamble and in new substep (c) to recite the language: "independently of a point of sale system". Parallel amendments have been made to independent system claim 8.

The Examiner states that Applicants further argue with respect to claims 1, 2, 5, 6, 12, 13, and 15 that Ogata, et al. fails to disclose an inclusion of point of sale equipment into the scope of the computer emulation for a target computer system and that the drivers of Ogata, et al. are those used to control devices used by the target system and not a point-of-sale device. The Examiner argues in turn, that "in combination with Barnstijn, et al., the feature making the target system a point-of-sale system is obvious because both Barnstijn, et al. and Ogata, et al. disclose systems which are geared toward the design and testing of applications to be implemented on a target machine, and Barnstijn, et al. does disclose that a target system can be a point-of-sale system (see claim 7).

Applicants respectfully disagree with the combination of these references by the Examiner. There is no motivation in Barnstijn, et al. or Ogata, et al. that would lead one to combine these references to arrive at Applicant's invention. Ogata, et al. teaches "an emulator that operates on an execution computer to execute programs that are designed to operate on a target computer". The target machine in this case is a "computer equipped with an architecture different from the architecture of the executing machine. The emulator comprises function modules that allow the executing machine to execute processes equivalent to processes that can be called via a jump table in the target machine. The emulator further comprises a transfer

controller that, when one of the processes in the target machine is called via said jump table, transfers the process to a corresponding one of said function modules in the executing machine prior to the actual onset of processing" (col. 2, ll. 15 – 36). This is not the same as Applicants' invention of providing an emulation module corresponding to a point of sale device. Claim 7 of Barnstijn, et al. does state that the target system can be a radio frequency point-of-sale terminal. The point of this claim however, is that the point-of-sale terminal can operate via an RF communication link between the host and target systems. The target system, even if it is a radio frequency point-of-sale terminal, still must include, as indicated in claim 5 of Barnstijn, et al.: (1) means for receiving said target input/output request signals transmitted by said host system via said communication link; (2) means for translating the target input/output request signals into target operating system calls which correspond to the operating system calls generated by the trial application program; and (3) means for executing the target operating system calls, thereby causing an interaction with the target input/output devices (col. 13, ll. 30 – 40). Applicants' invention requires no interface whatsoever with target input/output devices.

With respect to pending claim 7, it is Applicants' position that the combination of Barnstijn, et al. and Ogata, et al. fails to teach the steps of "providing an emulation object corresponding to the device", and "ensuring that the application would utilize the emulation object when the application is executed on the development system". The Examiner states that Ogata, et al. discloses the feature at column 5, lines 51 – 58, and that in combination with Barnstijn, et al., this feature is obvious because both references disclose systems which are geared toward the design and testing of applications to be implemented on a target machine. The Examiner further states that "Ogata, et al. disclose that the 'kernel' of the emulation program is loaded upon initialization. Ogata, et al. refers to the 'kernel' as a series of procedures within the

emulation process that ultimately determine why initialization took place and actually calls execution modules which perform emulation". The Examiner then refers to the Applicants' disclosure on page 10, lines 8 – 9, wherein there is a teaching that the emulation objects can be used to emulate the interaction between the application and specialized devices. The Examiner concludes that the "'kernel' of the emulation program and the emulation object of Applicants' invention both perform the same function simulating the emulation process". Applicants disagree with this conclusion.

Ogata, et al. disclose a computer emulator for emulating operating system function calls, basic input/output system (BIOS) function calls, basic disk operating system (BDOS) function calls, and interrupt tables of the target machine. As stated at column 5, lines 51 – 62: "the term 'kernel' refers to a series of procedures within the emulation processes that, when an exception handling is initiated, performs analysis to determine why the exception routine was initiated and calls execution modules which actually perform emulation via a process called a dispatcher. The term 'dispatcher' refers to a series of procedures within the emulation processes that call the execution modules which actually perform emulation based on the analysis by the kernel". The execution modules are depicted in Fig. 1 and are listed in column 11, lines 32 – 41. These emulators include a memory emulator, a mouse emulator, a graphic emulator, a text emulator, a character font emulator, a display emulator, a timer emulator, a keyboard emulator, and an interim controller emulator. All of these emulators represent execution modules under control of an operating system. There is no indication by the Examiner why one of ordinary skill in the art would use the Ogata, et al. reference with the Barnstijn, et al. reference in developing a method for testing an application on a development system that is completely independent of a point of sale system where the application will ultimately be executed. To more clearly distinguish the

present invention over the combination of Barnstijn, et al. and Ogata, et al., claim 7 has been modified to indicate in substep (c) that the application is executed on the development system independently of the point of sale system, and in the new substep (e) that the application is modified as necessary to ensure the application utilizes the emulation object on the development system.

With respect to claims 3, 4, 10 and 11, the Applicants' position is that Webber does not disclose platform independent emulation objects or specifically Java emulation objects representing a physical device. The Examiner states that the Webber invention concerns a merchant operated computer, configuring itself to access a production gateway computer, and does so by means of testing and development (configuration) and does disclose platform independent emulation objects or specifically Java emulation objects representing a physical device (see col. 7, ll. 11 – 17, col. 8, ll. 37 – 39). The Examiner states that Webber discloses that this invention may be implemented on several different platforms, including Java, which in turn means that the invention does not rely or depend on one single platform (platform independent). The Examiner further states that Webber discloses that this invention utilizes object oriented programming and objects, which can represent physical objects such as electrical components. Applicants disagree with the Examiner's conclusion.

At the lines cited by the Examiner, Webber teaches that his invention may be implemented on platforms and operating systems other than those mentioned. This is not a teaching that the application itself is platform independent. It simply indicates that the invention can be implemented on various platforms and operating systems. Webber further teaches that the preferred embodiment can be written using Java, C, and the C++ language and utilizes object

oriented methodology. Thus, there is no teaching in Webber that the emulation object is a Java emulation object.

New claims 16 and 17 are also submitted with this response after final rejection. The purpose of these additional claims is to include a further limitation on the step of ensuring that the application utilizes the emulation module when the application is executed on the development system. This limitation is the act of placing the emulation object in a class path that is higher than that of objects providing communications with the driver for the device. This means that when the application is running on the development system, the emulation module and the emulation object corresponding to the device will always be invoked.

It is respectfully requested that the Examiner fully consider this response after final rejection and enter the amendments to the pending claims and enter the new claims into the application. Furthermore, Applicants respectfully request that the Examiner reconsider and withdraw all outstanding rejections of the claims and allow the claims to issue.

Respectfully submitted,

John J. Timar

John J. Timar
Registration No.: 32,497

WOMBLE CARLYLE SANDRIDGE & RICE
P. O. Box 725388
Atlanta, Georgia 31139-9388
(404) 872-7000
(404) 888-7490 (fax)

CERTIFICATE OF MAILING

I hereby certify that this document is being deposited as First Class U.S. Mail in an envelope addressed to Assistant Commissioner for Patents, Washington, D.C. 20231 on November 22, 2000.

John J. Timar
Signature